

Exhibit D

Contemporaneous Document Written by Samuel Bankman-Fried

Case No. 22-cr-673 (LAK)

There were serious problems with the software development team at Alameda from November through March.

The beginning of the conflict was between Gary and [REDACTED]. They are both very talented developers, but have very different styles.

Gary's first instinct is to move quickly: to quickly get an intuition for what the desired outcome is, and straightforwardly build a system that delivers that. [REDACTED]
[REDACTED]

[REDACTED]

There were constant disputes between Gary and [REDACTED], and they often took the same form. Gary would have made some choice--maybe it was using Python, or some code design choice, or what type a variable should have. [REDACTED]
[REDACTED]

Eventually things further devolved. Gary built up most of Alameda's code base. [REDACTED]
[REDACTED]

Meanwhile, other developers started. They agreed that the code base was messy, but put in a bit of work and learned to work with it, quickly becoming valuable. Nishad and [REDACTED] spent a lot of work cleaning up various areas of the code, and implementing various helpful tools for the developers and the codebase.

[Nishad](#) in particular put in a herculean effort to fix the dev team. He took on all of the unrewarding code-cleanup tasks, dove in to the codebase--messy as it was--and tried as hard as he could to mend up dev team ego contests. He quickly emerged as the second most productive and valuable dev, despite his lack of experience, and as someone the trading team was excited to interface with.

[REDACTED]

[REDACTED]

[REDACTED]

[REDACTED]

[REDACTED]

[REDACTED]

[REDACTED]

The dev team culture degraded over time. Partially people got worn out of the infighting, mostly prompted by [REDACTED]. Partially meetings were unproductive; people would take up large amounts of time giving monologues on things that were dubiously right, and attempts to resolve disputes were unsuccessful. A large factor, too, was the lack of dev team leadership.

[REDACTED]

[REDACTED]

Gary was the default choice to manage the developers. He wrote half the codebase, has a great intuition for trading, is able to design and build large systems, and was the first developer. But he is also shy and not particularly interested in or motivated to manage people. He also really doesn't do well when he gets steamrolled in a conversation. He's quiet, and finds speaking out loud somewhat stressful--English isn't his first language, and he stumbles over words when he's thinking much more quickly than he can speak. He has a lisp that he's conscious of, and people who aren't putting in effort to understand him will often have some difficulty. And so when [REDACTED] would run over him in conversation, he would eventually give up on fighting back, and just quietly go on doing what he was going to do.

[REDACTED] saw himself as the natural choice to lead the dev team. He failed to ever get support from the developers for this, and it's easy to see why.

Nishad and [REDACTED] may have been able to handle the job socially, but didn't have enough dev experience to wield the technical respect necessary to get buy-in on their decisions.

There wasn't internal coherence or communication within the dev team, and there was no dev team manager to force people to cooperate and to resolve disputes. And so there was mass disorganization. Lots of attempts were made to use various task-management systems, but none really resolved the problems. Important tasks were handed to the dev team--like [keeping track of transfers](#)--and a few people would spend a few hours on them before giving up and leaving the tasks to languish in purgatory. It didn't help, also, that [REDACTED] strongly argued that we were not losing any transfers (we were); I think this probably didn't reflect a deeply held belief of his, but rather an instinctual reaction to argue against the position I took. Dev team [productivity](#) was quite low, and there was very little prioritization.

I do [deserve some blame](#) for this. I pushed heavily for us to get on more exchanges, which took up a bunch of dev time; I was over-optimistic about the amount we'd make on many of the exchanges. Even more than that, I could have taken a few actions to resolve the dev team power vacuum. I could have officially named Gary the head dev, and the dev whose word carried the ultimate decision making power even if he wasn't in charge of managing people. I also could have fired [REDACTED].

From very early on, I took Gary's side in the dev wars. I believed then, and still do now, that he did an extremely impressive job. He took in "we want to do good trades", and output a whole trading infrastructure in a few months. He made reasonable cost-benefit tradeoffs. And there were multiple instances of problems that seemed insurmountable that he solved in a few days--like trading on an exchange without an API by writing a chrome extension that mimicked a browser. Nishad nicknamed him "Gary the wizard", and that nickname really resonated with me, and with some others at Alameda.

took side.

and Gary also had trouble communicating with each other--partially just as a product of their other disagreements, but partially because they have very different communication styles. I'm guessing that Gary came across as robotic, dismissive, smelly, and silent to .

There were a lot of complicated debates at Alameda about which I'll never feel confident. This is not one of them. Gary almost single handedly built a trading firm's infrastructure in a few months, and then went on to build a functional crypto exchange in a few weeks. He was devoted to Alameda, willing to do whatever was most important for him to do, and had a knack for figuring out what systems people actually wanted.

Since the [employee exodus](#), Alameda has retained almost the entirety of the dev team output in Nishad and Gary while losing all of the drama. Gary has continued to unveil impressive projects at a startling pace, and [Nishad](#) has been taking increasing amounts of responsibility in a number of areas of the firm. Both of them are comfortable with roughly the entire codebase, bugs are much less common thanks to public goods like tests and documentation produced by Nishad, and interpersonal tensions within the dev team and spillovers to the company as a whole are gone.

In retrospect there are a lot of decisions I could have made about the dev team that would have been better than what happened. I could have appointed Nishad or Gary the dev team manager, despite the drawbacks. But, mostly, rather than try to hire our way out of the dev team mess (though did legitimately help), I should have addressed the source of the problem.